Scaling properties of on-line learning with momentum

Tom Heskes[1], Wim Wiegerinck[2], and Andrzej Komoda[2]

[1]Beckman Institute and Department of Physics, University of Illinois, 405 North Mathews Avenue, Urbana, Illinois 61801, U.S.A.

[2]Department of Medical Physics and Biophysics, University of Nijmegen, Geert Grooteplein 21, 6525 EZ Nijmegen, The Netherlands.

Abstract

We study on-line learning with momentum term for nonlinear learning rules. Through introduction of auxiliary variables, we show that the learning process can still be described by a first-order Markov process. For small learning parameters η and momentum parameters α close to 1 (we consider the case $\alpha = 1 - \sqrt{\eta/\lambda}$ for small η), Van Kampen's expansion can be applied in a straightforward manner. We obtain evolution equations for the average network state and the fluctuations around this average. These evolution equations depend (after rescaling of time and fluctuations) only on $\lambda = \eta/(1-\alpha)^2$: all combinations (η, α) with the same value of λ give rise to similar graphs. For small λ, i.e., $\eta \ll (1-\alpha)^2$, learning with momentum term is equivalent to learning without momentum term with rescaled learning parameter $\tilde{\eta} = \eta/(1-\alpha)$. Simulations with the nonlinear Oja learning rule confirm our theoretical results.

1 Introduction

Instead of the plain learning rule

$$\Delta w(n) \equiv w(n+1) - w(n) = \eta\, f(w(n), x)\,, \tag{1}$$

with $w(n)$ the network state at iteration step n (for notational convenience we will treat the weight vector w as a one-dimensional variable, generalization to higher dimensions is straightforward), x a randomly drawn training pattern, and η the learning parameter, we will consider the learning rule

$$\Delta w(n) = \eta\, f(w(n), x) + \alpha\, \Delta w(n-1)\,, \tag{2}$$

with α the momentum parameter. The incorporation of this momentum term is frequently applied to backpropagation with the intention to speed up learning (see e.g. [1] for references). The backpropagation learning rule is nonlinear in the weights w. Theoretical studies, however, have been mainly focussing on the linear LMS algorithm [2, 3]. In this paper we will consider the effect of the momentum term on nonlinear learning rules.

2 Doubling the system size

Equation (2) describes a second-order process. It can be turned into a first-order process through introduction of the auxiliary variable $\mu(n) \equiv \Delta w(n-1)$:

$$\begin{cases} \Delta w(n) = \eta\, f(w(n), x) + \alpha\, \mu(n) \\ \Delta \mu(n) = \eta\, f(w(n), x) + (\alpha - 1)\, \mu(n)\,. \end{cases}$$

0-7803-1901-X/94 $4.00 ©1994 IEEE

Defining $q \equiv (1-\alpha)\mu/\eta$, $\epsilon \equiv 1-\alpha$, and $\lambda \equiv \alpha\eta/(1-\alpha)^2$, we can rewrite this to

$$\begin{cases} \Delta w &= \lambda\epsilon\,[q + \epsilon\, f(w,x)/(1-\epsilon)] \\ \Delta q &= \epsilon\,[f(w,x) - q]\,. \end{cases} \tag{3}$$

We will study this system in the limit of very small ϵ and finite λ, i.e., for small learning parameters η and momentum parameters α close to 1. Note that if $\lambda = \mathcal{O}(1)$ the time scales of the equations for the weight w and the auxiliary variable q are of the same order. With finite $\epsilon = 1-\alpha$ and $\eta \to 0$, on the other hand, the evolution of the auxiliary variable q takes place on a much faster time scale than the evolution of the weight w. This situation, which requires a completely different analysis than the one presented in this paper, is treated in [4].

3 Van Kampen's expansion

The probability to be in a certain state (w,q) obeys a discrete-time random walk equation. This random-walk equation can be approximated for small parameters ϵ using Van Kampen's expansion [5, 6]. This expansion is based on the assumption that the stochastic process (3) can be viewed as a deterministic trajectory with (small) superimposed fluctuations. Starting from the Ansätze

$$w = \phi + \sqrt{\epsilon}\,\xi \qquad \text{and} \qquad q = \psi + \sqrt{\epsilon}\,\chi\,,$$

Van Kampen's expansion yields evolution equations for the deterministic variables ϕ and ψ, and for the average and (co)variance of the noise terms ξ and χ.

After rescaling time with $\lambda\epsilon$ (we define a new time $t \equiv \lambda\epsilon n$, where n is the time measured in number of learning steps), we obtain the deterministic equations

$$\begin{cases} \dot{\phi} &= \psi \\ \lambda\,\dot{\psi} &= f(\phi) - \psi\,. \end{cases}$$

with drift $f(\phi) \equiv \langle f(\phi,x)\rangle_\Omega$, where $\langle\cdot\rangle_\Omega$ denotes an average over the set Ω of training patterns x. The equivalent second-order differential equation is

$$\lambda\,\ddot{\phi} + \dot{\phi} - f(\phi) = 0\,. \tag{4}$$

The evolution of the averages of the noise terms follows

$$\lambda\,\frac{d}{dt}\begin{pmatrix} \langle\xi\rangle_\Xi \\ \langle\chi\rangle_\Xi \end{pmatrix} = -H(\phi)\begin{pmatrix} \langle\xi\rangle_\Xi \\ \langle\chi\rangle_\Xi \end{pmatrix},$$

where $\langle\cdot\rangle_\Xi$ stands for an average over an ensemble Ξ of learning networks and with

$$H(\phi) = \begin{pmatrix} 0 & -\lambda \\ -f'(\phi) & 1 \end{pmatrix}. \tag{5}$$

We will start our simulations with all learning networks at the same weight configuration, i.e., $w(0) = \phi(0)$ for all networks in the ensemble Ξ. This immediately implies $\langle\xi\rangle_\Xi = \langle\chi\rangle_\Xi = 0$ for all later times. The evolution of the average network state is thus completely described by the second-order differential equation (4).

With the following definitions for the covariance matrix Σ^2 and the diffusion matrix $D(\phi)$,

$$\Sigma^2 \equiv \begin{pmatrix} \langle \xi^2 \rangle_\Xi & \langle \xi\chi \rangle_\Xi \\ \langle \xi\chi \rangle_\Xi & \langle \chi^2 \rangle_\Xi \end{pmatrix}, \quad D(\phi) \equiv \begin{pmatrix} 0 & 0 \\ 0 & d(\phi) \end{pmatrix} \quad \text{and} \quad d(\phi) \equiv \left\langle f^2(\phi, x) \right\rangle_\Omega - \langle f(\phi, x) \rangle_\Omega^2 ,$$

the evolution equation for Σ^2 can be shown to obey

$$\lambda \frac{d}{dt} \Sigma^2 = -H(\phi)\, \Sigma^2 - \Sigma^2\, H(\phi) + D(\phi) . \tag{6}$$

The *a priori* Ansatz in Van Kampen's expansion is that the noise terms ξ and χ are of order 1. From (5) and (6), we see that this is valid for short times t and in regions of weight space where the real parts of the eigenvalues of the matrix $H(\phi)$ are positive, i.e., where $f'(\phi) < 0$. The same conditions hold for the validity of Van Kampen's expansion of the plain learning process (1) [6].

4 Scaling properties

With definitions

$$\sigma_1 \equiv \frac{1}{\lambda} \left\langle \xi^2 \right\rangle_\Xi , \quad \sigma_2 \equiv \frac{1}{\lambda} \langle \xi\chi \rangle_\Xi , \text{ and } \sigma_3 \equiv \left\langle \chi^2 \right\rangle_\Xi ,$$

the evolution equations for the average network state and for the fluctuations can be written

$$\begin{cases} \dot{\phi} - f(\phi) &= \lambda \ddot{\phi} \\ \dot{\sigma}_1 - 2\,\sigma_2 &= 0 \\ f'(\phi)\,\sigma_1 - \sigma_2 + \sigma_3 &= \lambda \dot{\sigma}_2 \\ -2\,\sigma_3 + d(\phi) &= \lambda \dot{\sigma}_3 - 2\lambda\, f'(\phi)\,\sigma_2 . \end{cases} \tag{7}$$

Note that (after rescaling of time and fluctuations) λ is the only remaining parameter in this set of coupled differential equations. Suppose we know, through calculations or simulations, $\phi(t)$ and $\sigma_1(t)$ for a particular value of $\lambda = \eta/(1-\alpha)^2$. Then for all combinations (η, α) with this particular λ the average weight and fluctuations at iteration step n follow from (recall our definitions of time t and variance σ_1)

$$\langle w \rangle (n) = \phi(\tilde{\eta}\, n) \quad \text{and} \quad \left\langle w^2 - \langle w \rangle^2 \right\rangle (n) = \tilde{\eta}\, \sigma_1(\tilde{\eta}\, n) ,$$

with "effective learning parameter" $\tilde{\eta} \equiv \eta/(1-\alpha)$. This effective learning parameter regulates the trade-off between speed and accuracy: a twice as large effective learning parameter leads to a twice as fast time scale, but also doubles the fluctuations in the weights. In the next section we will describe simulations with the nonlinear Oja learning rule to check these scaling properties.

For small λ we can further simplify the set of equations (7). As in [4] there are two different time scales: a slow time scale for the evolution of σ_1 and a fast time scale for the evolution of σ_2 and σ_3. If we neglect all terms of order λ, we can eliminate the variables σ_2 and σ_3 and find

$$\begin{cases} \dot{\phi} &= f(\phi) \\ \dot{\sigma}_1 &= 2\, f'(\phi)\, \sigma_1 + d(\phi) . \end{cases}$$

The same set of equations is obtained if Van Kampen's expansion is applied on the plain learning rule (1) with effective learning parameter $\tilde{\eta} = \eta/(1-\alpha)$ (see e.g. [6]). Similar results have been reported in earlier studies on linear learning rules [2, 3] and nonlinear learning rules [4].

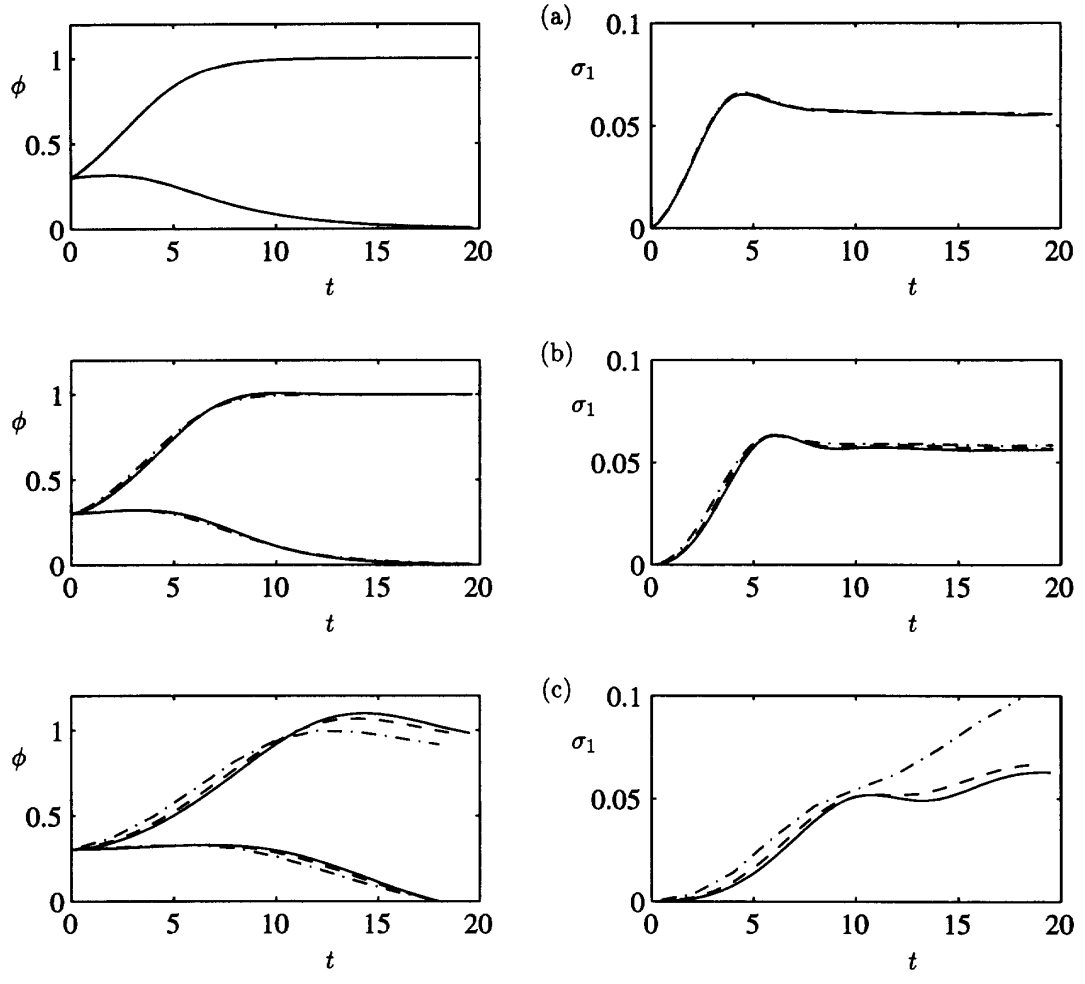

Figure 1: Oja learning with momentum updating. Means ϕ and (rescaled) sum of variances σ_1 as a function of rescaled time t. All 100 000 networks started at $w = (0.3, 0.3)^T$. Momentum parameter $\alpha = 0.9$ for the solid lines, $\alpha = 0.8$ for the dashed lines, and $\alpha = 0.6$ for the dash-dotted lines. (a) $\lambda = 0.1$; (b) $\lambda = 1$; (c) $\lambda = 5$. The close correspondence between curves with constant λ for small learning parameters η and α close to 1, confirms the theoretically derived scaling properties.

5 Simulations

The nonlinear Oja learning rule [7]

$$\Delta w \;=\; \eta\,(w^T x)\,[x \;-\; (w^T x)\,w]$$

searches for the principal component of the input covariance matrix $\left\langle x\,x^T \right\rangle_\Omega$. Inputs are drawn at random from a two-dimensional rectangle centered at the origin, with sides of length 2 and 1 along the x_1- and x_2-axis, respectively. Simulations are performed with an ensemble of 100 000 independently learning networks, all starting at $w(0) = (0.3, 0.3)^T$. Since the principal component

lies along the longest side of the rectangle, the weights w_1 and w_2 tend to 1 and 0, respectively. Figure 4 shows the evolution of the average weights and of the trace of the covariance matrix for different values of α and η such that $\lambda = 0.1$, 1 and 5 (figure 4(a), (b), and (c), respectively). Time and variance are rescaled with $\eta/(1-\alpha)$, i.e., we plot ϕ and σ_1 as a function of the rescaled time t. Curves with constant λ are almost overlapping, except for the quite extreme values $\alpha = 0.6$ and $\eta = 0.8$ [dash-dotted lines in figure 4(c)]. These simulation results are in perfect agreement with the theoretically derived scaling properties that should be valid for small learning parameters η and momentum parameters α close to 1.

6 Summary and discussion

In this paper we studied nonlinear on-line learning rules with momentum term. Using Van Kampen's expansion we derived evolution equations for the average network state and the fluctuations around this average. Strictly speaking, these equations are only valid for small learning parameters η and momentum parameters α close to 1. Simulations indicate that these evolution equations can be accurate even for relatively small momentum parameters α. For combinations (η, α) such that $\lambda = \eta/(1-\alpha)^2 \ll 1$, learning with momentum term is equivalent to "plain" learning with rescaled learning parameter $\tilde{\eta} = \eta/(1-\alpha)$. We obtained this result as a special limit of the more general set of equations (7). Further study of this set of equations for finite λ should reveal whether incorporation of the momentum term leads to a better performance of nonlinear on-line learning rules.

Acknowledgments

This work was supported by the Dutch Foundation for Neural Networks (WW and AK) and by a grant (P41RR05969) from the National Institutes of Health (TH).

References

[1] J. Hertz, A. Krogh, and R. Palmer. *Introduction to the theory of neural computation.* Addison-Wesley, Redwood City, 1991.

[2] J. Shynk and S. Roy. The LMS algorithm with momentum updating. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, pages 2651–2654, 1988.

[3] M. Tugay and Y. Tanik. Properties of the momentum LMS algorithm. *Signal Processing*, 18:117–127, 1989.

[4] W. Wiegerinck, A. Komoda, and T. Heskes. On-line learning with momentum for nonlinear learning rules. *Submitted to ICANN'94*, 1993.

[5] N. van Kampen. *Stochastic processes in physics and chemistry.* North-Holland, Amsterdam, 1992.

[6] T. Heskes. On Fokker-Planck approximations of on-line learning processes. *Submitted to Journal of Physics A*, 1993.

[7] E. Oja. A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–273, 1982.